Dr Jonathan Hare G1EXG
jphcreativescience@gmail.com

In my last article we looked at how AI helped design a large air cored coil for VLF work. We now look at using it to help us design the CI-V data frames needed to control an Icom IC-7300 (MK I or MK II) or IC-9700 transceiver. As with part 1, I will continue to use a 'running score' after each section or paragraph, as a way of expressing if I thought my overall AI experience was helpful (✓) or unhelpful (X).

Using the CI-V interface on IC7300 & IC9700 transceivers

I have been wanting to create a project to request S-meter readings from my IC-9700 and IC-7300 transceivers to log beacons and for antenna tests etc. I used the tiny one board computer, the Pi Pico to do this. The Pico can also generate on-board CSV files to store S-meter values for evaluation and analysis later by a spreadsheet. I had made an interface board two years ago, but I never had the time to get started properly on the coding [**1**].

A second project that came up was for the Worthing and District Amateur Radio Club (WADARC). They have a networked IC-7300 MK II that members can access to use the radio remotely. The radio can transmit from topband to 4m so it is important to make sure the correct antenna is selected at all times. If 100 watts on 7MHz was accidently fed directly to the 4m antenna it could easily damage the output stage of the transceiver or the antenna. One way is to get the pi Pico to request the frequency and then use the Picos's output pins to switch RF relays, so the correct antenna is always selected [1].

As we will see ChatGPT really helped me overcome the barriers to start the coding for these projects and even though it sometimes provided incorrect answers, I doubt I would have got the project going so quickly without it.

The full IC-7300 and IC-9700 manuals are available as PDF downloads from the Icom website. These explain exactly what CI-V frames you need to send the transceivers to perform various actions remotely. However, converting what you read there into a working project requires a bit of time and patience and ChatGPT really helped.

Introduction

When it comes to programming, I am what you might call an experienced beginner. I tend to get involved in a coding project for a few months then not touch coding again for months or even a year. Although I am quite good at problem solving and getting things to work, I am poor at spelling and remembering the correct code syntax. In the months between coding sessions, I often struggle to remember if 'adc.read_u16()' is correct, when to use capitals, are the brackets in an array square or curved? When do you use a single '=' or double '==' etc. etc.

There are some very good books that can help with understanding both the Pi Pico hardware and Python coding which I make use of extensively [**2**, **3**], but I have also found ChatGPT to be very useful. It seems to be very good at reading my terrible spelling and usually knows what I am trying to ask. It is also expert at generating code with the correct syntax (even if the code might not work … more later).

The result is that I can have the ideas while I can use ChatGPT to help me with the details. It is a good combination, as even if it does get things wrong, I can usually spot these sort of mistakes and work around them. The result is that I really do seem to be able to make progress fast with the help of AI. Using it I can see progress unfolding.

ChatGPT and computer code

Unlike our spoken language, programming languages are very well structured and self-consistent. AI chat bots such as ChatGPT or Claude can generate working code very effectively – providing of course you ask the right question with as much detail as possible. For example, in the Pi Pico S-meter project (see below) I might ask:

"write code to receive CI-V s-meter data on an Icom IC7300 or IC9700 transceiver"

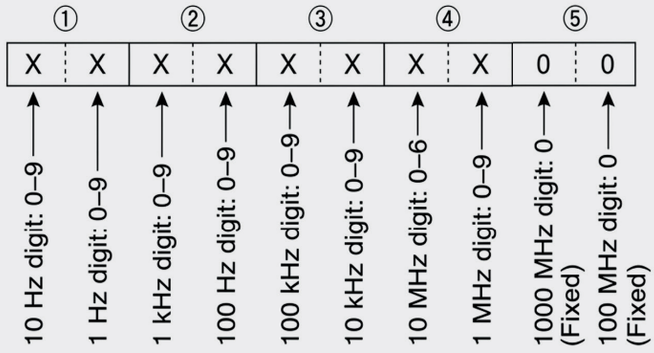
But you might find you get code written for an Arduino (so not in Python) and even if you do get

Remote control (CI-V) information (Continued)

◇ Data content description

• Operating frequency

Command: 00, 03, 05, 1C 03



• Operating mode

Command: 01, 04, 06

Filter setting (②) can be skipped with command 01 and 06. In that case, "FIL1" is selected with command 01 and the default filter setting of the operating mode is automatically selected with command 06.

1

Using AI Assistants such as Chat GPT for Radio Projects (Pt II)

Dr Jonathan Hare G1EXG completes his overview of using AI in amateur radio projects.

Sign up to our FREE email newsletter at www.mymagazinesub.co.uk/practical-wireless

Chat GPT.indd 48 08/06/2026 08:58

Fig. 1: Coding of frequency information for IC-7300. Fig. 2: Suggested (ChatGPT) interface.

code written in Python, it might be for a Ras Pi rather than micro python required for the Pi Pico.

So, it is better to include as much detail as possible e.g.:

"write micro python code to run on a Pi Pico (19200 baud) to send out CI-V data on the UART0 to receive s-meter readings from an Icom IC7300 or IC9700 transceiver".

I did this and immediately got code that worked, although as we will see not without some problems.

Cumulative Score
helpful: ✓✓✓✓✓✓ unhelpful: XXXXX

More often the free version of ChatGPT provides code that does not work straight away. However, you can type in the errors you get running the code and ask ChatGPT to make changes or corrections. Saying that, it sometimes seems to get into a real mess trying to correct things. If you find you seem to be going around and around with no progress, it may be better to stop the current chat stream and simply start again with a new prompt with what you now know works.

Cumulative Score
helpful: ✓✓✓✓✓✓ unhelpful: XXXXXX

CI-V frames

Both the IC-7300 (MK I and II) and IC-9700 use an Icom system of communication – the CI-V 'one wire' interface - between the radio and a computer or microcontroller device. This can be done three ways via the connections on the back of the radios: i) a network connection (IC-7300 MK II), ii) USB or iii) a 3.5mm mono jack socket. For these experiments I will be using the two-wire arrangement via the 3.5mm mono jack.

You can't directly connect a Pi Pico to the transceiver; you need a simple interface. We will discuss this later in the article, but first let us look at what data – 'the frames' – needs to be sent to do something

Changing the power

Let's discuss what frames are required to change the output power of the radio. A frame consists of a series of hexadecimal (hex, 0x) numbers sent via a UART along the CI-V data line to the radio.

All the frames start with two start bytes 'FE', then the address of the Icom radio (IC-7300 = 94) is sent, then the address of the sending microcontroller (the Pi Pico, E0), then one or two bytes are sent corresponding to the particular function you want to change (in this case 14 followed by 0A), then you send the data you want to use to change the setting on the radio and finally send the end of frame byte, 'FD'.

The transmit power can be changed from 0% (0 watts) to 100% (e.g. 100 watts or ca. 50 watts on 70MHz). The two data bytes are sent Binary Codded Decimal (BCD) and form a number between 0 to 255. 0 being 0% (0 watt) and 255 being 100% (100W) power. The numbers are sent as hex but are always in the range 0-9 like decimal numbers (i.e. despite being hex the data never contains A, B … or F's).

To set it to 100% (100 watts) you need to send '255' in the two data bytes. It's made up of BCD so the most significant byte (MSB) will be the 100's i.e. 2 while the least significant byte (LSB) will be the 55. So, you send "02" and then "55" in hex for the data bytes. Saying out loud *"two hundred and fifty five"* divides it up a bit like BCD.

If you want to change the power to 5% (5 watts) the data value needs to be 5% of 100 so 5/100 x 255 = 12.75, so about 13. So, for 5 watts you would need to send "00" and then "13".,

Here are some examples:

5% Tx power send: FE,FE,94,E0,14,0A,00,13,FD

50% Tx power send: FE,FE,94,E0,14,0A,01,28,FD

100% Tx power send: FE,FE,94,E0,14,0A,02,55,FD

How did ChatGPT get on creating code to get this to work?

The first time I asked ChatGPT to work on this problem it assumed the data was one byte (presumably because the data value was in the 0-255 range), so this failed to work. When I reminded it that the data was in two bytes it converted the data into a 16-bit number and tried to get that to work but failed of course. Finally, when I gave it a few more details saying the data needed to be two bytes BCD it did create code that worked [1].

Cumulative Score
helpful: ✓✓✓✓✓✓ unhelpful: ✗✗✗✗✗✗✗

WADARC frequency request

In the WADARC project we need to request the frequency from the radio [1]. In this case you send the appropriate frame but this time we also need to get a reply frame back that contains the frequency data (in BCD). **Fig. 1**, a diagram from the Icom IC-7300 PDF data sheet, shows how the frequency information is coded.

How did ChatGPT do?

Firstly, as discussed above we need the prompt to remind ChatGPT that the data is in Binary Coded Decimal. Even so AI got into all sorts of difficulties. I think partly perhaps because when you take the frequency and divide it up into the 5 bytes you i) need to add zeros to get the in the correct format, ii) you need to reverse the order and iii) you also need to make sure the least significant nibble of each byte also gets sent in the right order.

After you send the request frames you get back the following frames:

FE FE E0 94 aa bb cc dd ee FD

Schematic (ASCII)

```
|
    Icom IC-7300 / IC-9700
     3.5 mm CI-V jack
       TIP
        |
       100Ω  (optional)
        |
        +──────────────┐
        |              |
        |             C|
        |      Q1     |  NPN (2N3904)
        |             E|
        |              |
        |              +──────────── GND
        |
        +──────┬──────────── Pico GPIO (CI-V)
               |
              10k
               |
             3.3 V
```

Q1 base connection:

```
Pico GPIO (TX control)
  |
 4.7k
  |
 B (Q1)
```

Grounds must be common between the radio and the Pico.

2

Buy back issues and archive CDs at www.mymagazinesub.co.uk/practical-wireless

Chat GPT.indd 49 08/06/2026 08:58

Where aa to ee are the hex BCD frequency bytes

As an example, if we tune the radio to 14.070MHz, what frames we should get back for this frequency?

First, we add zeros both sides so 14.070MHz becomes 0014.070000MHz (i.e. to 10Hz res.)

(Note: only the IC-9700 uses the first set of bytes as it needs them to show 1GHz and 100MHz for the 23cm band etc.)

Then we break this up into pairs: 00 14 07 00 00

We now run the pairs in reverse order to get the final correct layout of the five pairs of frequency bytes: 00 00 07 14 00.

Finally, this is the frame we should receive on requesting a radio set to 14.070MHz:

FE FE E0 94 00 00 07 14 00 FD (btw I measured this directly from the radio to check)

How well did ChatGPT cope with this challenge? Unfortunately, it kept saying the frequency bytes should be: 00 00 70 40 01 and claiming the IC-7300 frames received back should be: FE FE E0 94 04 00 00 70 40 01 FD

Cumulative Score

helpful: ✓✓✓✓✓✓✓✓ **unhelpful:** ✗✗✗✗✗✗✗✗✗

I tried other frequencies just in case it was digging up an example that was wrong. I tried again a few days later on some calculations. It got part of the five bytes calculation correct but I found it sometimes mucked up the middle ones, almost as though it got distracted by something along the way!

Cumulative Score

helpful: ✓✓✓✓✓✓✓✓ **unhelpful:** ✗✗✗✗✗✗✗✗✗✗

For my WADARC project I just need to know the integer MHz value i.e. if on 3.755MHz I simply need to get 3, if on 14.250MHz I get 14 etc. It turns out this data is all in the fourth byte. The MSB contains the 10's MHz values, while the LSB contains the 1's MHz values. So, to 'calculate' what we need for the14.250MHz example the MSB of byte four will give me the 10's MHz while the LSB will give me the 1's MHz. I then get this from byte 4 and multiply the 1 x 10 and add the 4 to get 14.

After much toing and froing ChatGPT did provide quite a robust code that not only sent the correct frames but also checked the received frames to ignore other CI-V traffic. I would not have done this on my first attempt, so this was good.

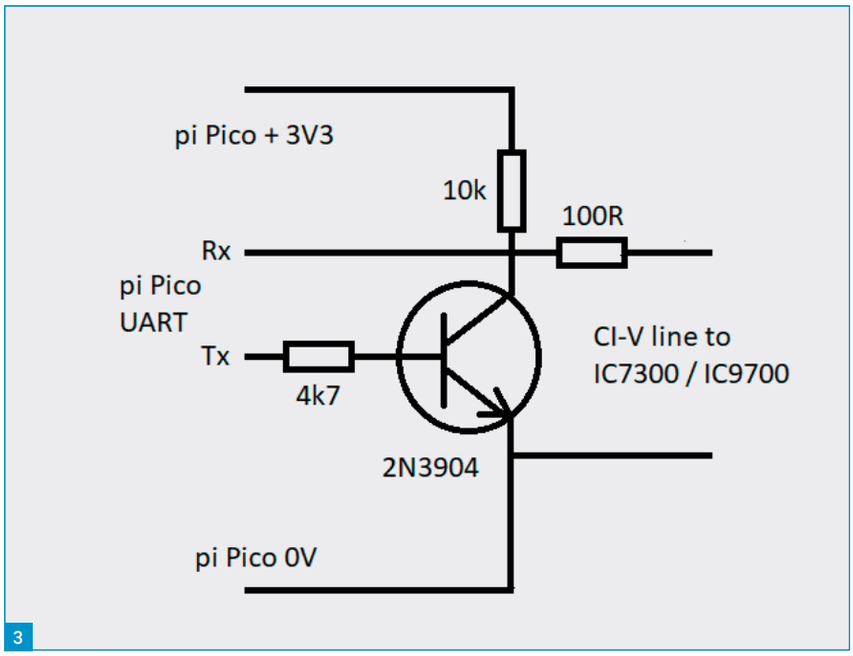
But ChatGPT made a complete mess of deciding which of the bytes I would need to use. It often suggested I would need the MSB nibble of one byte and the LSB of another. I never did get it to provide proper working code, but it did get me a long way and basically told me everything I needed to know for me to work it out for myself! [1].

Cumulative Score

helpful: ✓✓✓✓✓✓✓ **unhelpful:** ✗✗✗✗✗✗✗✗✗✗

Interface design

ChatGPT is not so good designing diagrams and circuits - not yet anyway. For example, I asked it for help designing an interface to go between the Pi Pico and the Icom transceiver. When asking about this I mistakenly said 'the 2.5mm socket on the back of the radios'. It did not pick up that it is actually a 3.5mm mono jack socket. It did say however, that the best interface was to use an opto-isolator so that the radio and microcontroller are electrically isolated from each other.



3

However, it kept suggesting a simple one transistor interface to try first. The ChatGPT 'circuit' for this is shown in **Fig. 2** created using ASCII symbols which I have also redrawn as a standard circuit diagram, **Fig. 3**. There are many problems with this simple circuit, but one is that a typical UART idles high when no data is sent. The inverting action of this simple circuit means the CI-V will idle low – i.e. be pulled low / down and this will basically hijack / dominate the CI-V line and not allow other CI-V traffic or data to come back from the transceivers.

Cumulative Score

helpful: ✓✓✓✓✓✓✓✓✓

unhelpful: ✗✗✗✗✗✗✗✗✗✗✗✗✗

ChatGPT got so muddled I gave up on trying to use it to design the interface and opted for one published by W8WWV [**4**]. His Icom interface uses a 7417 TTL buffer / line driver and works well. I ran it from the Pi Pico's 5V line but modified it with a simple two resistor potential divider, so that the RX signal could go into the 3V Pi Pico UART input (see **Fig. 4**) [1].

Other miscellaneous electrical and radio topics

Here are a few radio and electronics AI investigations that I am doing that are still on-going:

23cm: ChatGPT has come up with some fascinating comments about the role water plays in propagation at 23cm which I have not seen before. It suggested that water vapour in the air has relatively little effect on propagation at 23cm but that wet objects along the path (house, trees, land) can cause large attenuation of the signals.

Cumulative Score

helpful: ✓✓✓✓✓✓✓✓✓✓✓

unhelpful: ✗✗✗✗✗✗✗✗✗✗✗✗✗

4:1 half-wave coax balun: If you ask what the voltage is midway along the halfwave part of a 4:1 coax balun (as used on many 2m band Yagi designs for example), you get that it is voltage maximum not minimum. I didn't expect this and reminded me that I needed to think not only about both the voltage and phase along the device, but that it is also that the RF is a wave moving along the device. I am still not sure I understand this, but it is thought provoking.

Cumulative Score

helpful: ✓✓✓✓✓✓✓✓✓✓✓✓

unhelpful: ✗✗✗✗✗✗✗✗✗✗✗✗✗

Synth gens: ChatGPT seems to be fairly good at calculating register values for setting up synthesiser chips such as the ADF4351 which can generate 40MHz to 4GHz. But obviously the 'proof is in the pudding' – you need to try them out. You can ask for low phase noise choices etc. Add as much information in the prompt as possible.

Cumulative Score

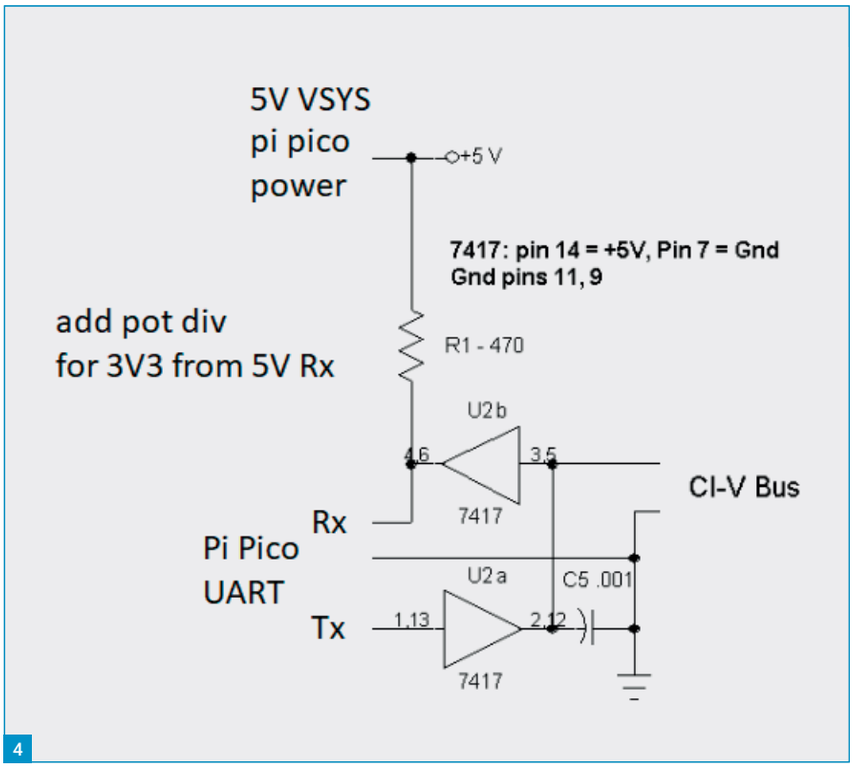
helpful: ✓✓✓✓✓✓✓✓✓✓✓✓✓

unhelpful: ✗✗✗✗✗✗✗✗✗✗✗✗✗

Sensors and peripherals: I often make use of 2 or 4-line LCD screens (4-wire interface) in my microcontroller projects. ChatGPT was helpful getting me started using the compact and versatile OLED displays to work with a pi Pico.

Sign up to our FREE email newsletter at www.mymagazinesub.co.uk/practical-wireless

Chat GPT.indd 50 08/06/2026 08:58



Cumulative Score
helpful: ✓✓✓✓✓✓✓✓✓✓✓✓✓
unhelpful: ✗✗✗✗✗✗✗✗✗✗✗✗✗
Helix antenna design: ask ChatGPT *"In microwave helix antennas a small triangular section of copper is often used to transform the ca. 110 ohm helix impedance down to 50 ohm, how does it work and how long does it need to be?"* and it produced a well written and helpful account of how this 'planar impedance taper' works.
Cumulative Score
helpful: ✓✓✓✓✓✓✓✓✓✓✓✓✓
unhelpful: ✗✗✗✗✗✗✗✗✗✗✗✗✗

Summary

AI chat bots such as ChatGPT offer a fantastic way of exploring projects and research ideas provided you always check important facts. A few years ago, you had to treat AI rather like you might treat meeting a slightly drunk expert as it produced garbled, rambling hallucinations and simply making things up. But on the whole, provided you ask the right question, it now seems to be much better – at least for technical enquires.

Not surprisingly (for a language-based system) AI still seems to excel at textual based Q&A, including coding, but at time of writing it often fails spectacularly (and quite frankly idiotic) when trying to create diagrams or graphs etc. When it does this ChatGPT seems less like a technological wonder and more like *Private Eye's* funny teaser *"Chat Bot PG Tips"!* [5]

AI chat bots can sometimes get stuck in a sort of righteous loop, claiming something is correct when it clearly is not. In that case it worth stopping and thinking about a way in which you can start off with information that you know is correct and use this as a base for a new prompt to guide it.

The advantage of AI over a human is that you can enquire over a much wider range of topics and expertise, and it never gets bored, frustrated or irritated by your questioning. Further, if something out of the blue occurs to you, you can easily go off topic without AI raising an electronic eyebrow.

I am quite good at getting things to work once I have something to work with. My stumbling blocks or 'barriers to creativity' are often getting up to speed with the data and information required to get a new project or experiment going. I find that I can use ChatGPT to rapidly get me started on projects and then once going I can cope with the mistakes and confusions AI can sometimes create along the way.

My very un-scientific total score in these projects was that ChatGPT gave 16 'helpful' and 14 'unhelpful' scores - overall I guess that equates to *"ok, but could do better"*. But these are very early days, and I am using the free version of ChatGPT. Other chat bots (such as Claude) may do better.

I have found my investigations using AI were usually fun, inspiring and only a little frustrating or worrying. Going back to the **Harry Kroto** quote, asking the right question is key – what in the industry is called 'effective prompt engineering'.

If I had any suggestion how AI bots such as ChatGPT might be improved, it would be that instead of diving straight in creating apparently 'confident' answers, that it might instead, pause to see if it needs to ask for more fundamental information. This would help us to create better prompts leading to better answers. Of course, this all rather assumes that AI is intelligent which of course it is not. That's why I prefer to call chat bots 'digital assistants' rather than use the term 'artificial intelligence'.

These are early days for AI, and there is no doubt that the tremendous rate of development will result in dramatic improvements – and probably very soon.

And finally...

If you want to see just how amazing exploring radio and 'the whole universe' can be with AI try typing in the following prompts, you might be surprised:

"A wave breaks because the water closer to the sea floor moves slower, can a similar effect take place with a radio wavefront?"

And while you are at it also try:

"If a VLF transmitter is creating a vertically polarised signal, will the polarisation at some lower latitude appear horizontal polarised?"

Acknowledgements

I would like to recognise the late Sir Harry Kroto, **Dr Richard Robinson** (Brighton Science Festival), **Dr Bernd Eggen**, **Dr Chris Ewels**, **Sarah Hogben M6GDW**, **Andrew Pratt**, **Simon G4EYR**, **Roger M0TJK** and **Graham G8BZL** for many useful discussions. Finally, I didn't use ChatGPT to edit or help me write this article, so any errors are presumably my own. **PW**

References and links

[1] CSC Pi Pico web page: **www.creative-science.org.uk/pi_pico.html**
[2] Get Started with MicroPython on Raspberry Pi Pico Book Gareth Halfacree, Ben Everard Raspberry Pi Press ISBN 978-1-912047-86-4
[3] Raspberry Pi Pico Essential, D Ibrahim. Elektor book, 2021. ISBN 978 3 89576 427 1
[4] search for "W8WWV Icom interface using 7417 buffer / line driver"
[5] Private Eye Annual 2025

Fig. 3: **Fig 2 redrawn as a more familiar circuit.**
Fig. 4: **Modified W8WWV interface.**

Buy back issues and archive CDs at www.mymagazinesub.co.uk/practical-wireless

Chat GPT.indd 51 08/06/2026 08:58